Platform for intelligent management of Industrial Machinery based on service-oriented architecture

Luis Felipe Herrera-Quintero, Vicente Berenguer-Miralles, Felipe Restrepo-Calle,
Raúl Gómez, Virgilio Gilart-Iglesias, Francisco Maciá-Pérez.
Technology and Computer Science Department
University of Alicante
PO Box 99, 03080, Alicante
SPAIN
http://www.dtic.ua.es/grupoM

Abstract: - In this paper, we present a platform designed and implemented for the practical management of industrial machinery, which provides through an organization orientated to services the processes of selfmanagement and proactive management of industrial machinery. This platform allows to establish several relationships with the services offered by the machine, which makes possible that the machines can communicate and cooperate between itself or with other elements of production. The platform that was made, is based on service-oriented architecture (SOA) and Web services in order to obtain that the industrial machine provide high levels of selfmanagement and overcome the barriers imposed by the physical and technological restrictions which prevent to these components achieve the transparent integration of industrial machine with the business processes of the organization.

Key-Words: - **Embedded Systems, Agile Manufacturing, Business Processes Management, BPMS, BPEL.**

1 Introduction

Nowadays most of the manufacturing processes are defined of way ad-hoc in order to the type of activity which conform and the type of resultant product. For example, inside of a car's assembly a robot arm has several tools and its function is performed on the assembly line of vehicle chassis. The robot arm will require a programming completely dependent on other machines in the chain, and if they wanted to reuse the robot arm in another process is necessary reprogramming itself to adapt to the new environment, so that the costs associated with such processes will be absolutely highest.

This paper presents a platform that automates the integration of Information Society Technologies (ITS) in industrial environments using Web services based on the paradigm of service-oriented architecture (SOA). In this way the industrial machinery can offer his functionalities through services that may be published, discovered and consumed by other elements of the system in automatic way [1]. To this we propose introducing a embedded system in each one of the machines, which provide a standard interface for access independent of protocol used in industry.

The great advantage of this environment is flexibility, because each machine connected to the network will provide a range of services and will be able to reconfigured new processes and activities to suit a certain environment without need for reprogramming a low level. All this leads to the machine is abstracted from the characteristics of low level and it will be possible to interact with it in an environment more functional.

There are also a number of value-added services that allows incorporate ITS infrastructures in industrial machine such as how to control or monitor the various processes. An example would be obtaining statistics or historical data from various processes [2]. This improves not only the process locally but globally the enterprise production system.

We must also take into account that the incorporation of Web technologies, as the TCP/IP communication model allows decoupling of the physical tasks of production and management [3], so that we can monitor and manage our business process since anywhere and anytime. Incorporating this level of ubiquity will make a noticeable difference with the traditional industrial manufacturing environments.

This article is organized as follows: In section two we will give a few details about the background and state of the art inherent in the proposal, in section three we propose a general scheme about operating the platform, and as their characteristics and requirements, section four focuses on the implementation and integration within a real scenario for industrial simulation, in section five are carried out tests of the platform and section six we concludes and we describe improvements over traditional

systems and we will propose which are the lines that we will work.

2 Background and state of art

The growth of the ITS together with Internet evolution have done that several organizations change their business strategies, redefining their current paradigms, in order to continue being competitive in the current market. In this way, and focusing in the use of the eBusiness technologies in the industries, the new technologies platforms have to include distributed software components that allow increasing the organization competitiveness levels [4]. However, in order to achieve a transparent integration of these components is necessary to identify the existent lacks in the industrial machinery environment.

For a long time, several protocols and technologies in order to achieve the integration of the industrial machinery have been developed following the traditional automation models [1]. In this way, the development of owner protocols such as Modbus, Profibus, DeviceNet, Industrial Ethernet produce several devices that allow automating a great set of industrial processes. Nevertheless, these approaches differ of the business model presented in this paper. Its main goal is to introduce the TIC technologies in order to achieve models with high level of customization [5].

It is here where several lacks have been identified, and for this reason, organization as ABB or Schneider, organizations leaders in automation processes, propose the incorporation of the embedded systems that allow an integral management of their production devices using convectional network models such as TCP/IP [6]. Also, they propose to introduce higher IT paradigms like Service Oriented Architecture using SOAP protocol [7] to establish the communication process between different production devices.

There are several research and development European projects inside of the ITEA program [8] that have similar goals that the approach described in this paper. This is the case of the SIRENA project [9] whose main aim is the creation of a framework that allow the specification and development of distributed applications for real time embedded systems for industrial automation and automotion industry.

There are different technologies, for examples Web Services, which support the presented approaches in order to create a platform that allow to integrate different industrial machinery processes in the organization business models.

This involves the use of several technologies such as WSDL [10], SOAP, UDDI [11] and BPEL [12] that provide more flexibility in the platform design. The Web Services Description Language (WSDL) [13] is a specification created in order to describe and publish in a standard way the Web Service invocation formats and protocols. This standard to describe Web Service Interfaces is necessary to avoid creating special interactions with each service provider in the network. WSDL is based on XML language [14], and include data description that has to be sent to the Web Service.

Other important technology is the Business Process Execution Language (BPEL). Basically, we use this language to control in a centralized way the invocation of the different Web Services. This language includes certain business logic that makes easy the Web Service composition process. That is, a definition of a BPEL process use one or more web Services described by WSDL interface. In addition, provide a behavior description and the interactions of a process instance concerning its components and resources through these interfaces.

3 Overview of the proposal

As discussed in the introduction, in our design we can establish a correspondence between a Web service and an industrial machine with a defined standard interface. So we can define a service-oriented architecture within an industrial environment that consists of three modules. The first one includes the Industrial Machinery and embedded device that makes it possible integration with Web technologies, the second one module is a register where industrial machine advertise their services, and the third module consists of the process management system (Business Process Management System: BPMS) where different processes are arranged to be executed later in the industry machinery.

In the Fig 1 we can see how the modules interact through a series of steps: publication of machine services, search services, discovery of services and consumption of services. These steps will be explained in depth in implementation section. Now we will focus on the design of each of the modules that make up the platform.

The first module, called *IMaaS* (Industrial Machines as a Service), comprises the different industrial machines that provide physical services. This module includes also an intermediate element *WS Gateway* (Web Service Gateway) as a gateway between the machine and the network. It is responsible for publishing services and meeting the requests on them.

The second module (Service Broker) called "UDDI / Service Register" is responsible for registering all services offered by each machine in the industrial environment. This allows adding intelligence to our proposal, because if we know the function of each machine and it is in a centralized registration system, it is possible balancing the load of the industrial environment to scalar levels. When a client (consumer) wants to use a service, the client makes a search in the UDDI registry and it provides the needed data to communicate with the machine. With this data the client will know the structure of machine and may proceed to consume their services.

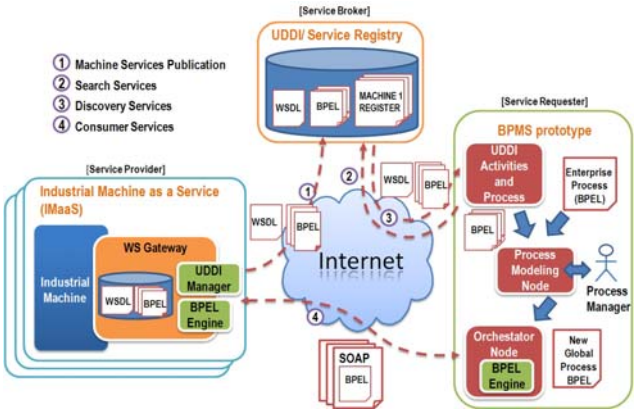

Fig 1. General scheme of the proposed system

The third module, called BPMS or Service requester, is a central management system that is responsible for accessing the UDDI registry to obtain the services it provides each component of the industrial environment. From activities and processes obtained and a series of custom processes that keeps the company, it is modeled a global process that will be executed by the node orchestrator's BPEL engine.

The normal pattern of activity in a scenario like the one proposed in figure can be summarized in the following phases:

1. Registration of machinery against UDDI.
2. Search UDDI.
3. Modeling processes.
4. Executing process.
5. BPEL engine. Execution of local processes.

In phase 1, when the machines are connected to the network, the first thing they should do is automatically register their services in the UDDI registry available in one of the nodes in the system.

An issue that we must take into account is how to represent the offered service by each machine and describe each of the activities and processes that compound this service. We will use a different WSDL and BPEL sheets describing each of the processes that is capable of carrying out the machine and once described thus services, we must determine how to register these in UDDI, with the goal of the orchestrator node can do parameterized searches, learn the characteristics and references to the various services and compose global processes. To carry out the correspondence between BPEL and UDDI we have followed the recommendation made in [15] and [16].

In phase 2 are used functions to question or search, with the aim of extracting a list of services available on the assembly chain and operations (activities and processes) they offer. Also, from the central node system, it is possible to contemplate quickly, clearly and structured the components of the chain and the roles they can play. The platform offers targeted searches in the UDDI registry of any machine, a particular service, activities or processes with specific features, and so on.

Once you get the list of services connected to the network of the assembly chain with their respective operations is possible modeling a global BPEL process (phase 3). In addition to the services stored in the UDDI registry, it employs a number of predesigned and custom processes, which may be used as threads of commented overall process.

Once composed the overall process with a subset of operations available on the assembly chain, it's possible to run it (phase 4). To do this we need a planner to analyze the overall BPEL process and go running as a batch file. For each step of the process (activity, process or predesigned process) the planner will generate a SOAP message with the corresponding BPEL code and sends it to the machine responsible for the execution. The structure of this message can be seen in Fig 2.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlnsoap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<process ....>.....
<sequence name="Sequence1">
<invoke name="Invoke1" partnerLink="Move"
operation="ArrancarDerecha"
portType="ns0:InterfazCliente"/>
</sequence>
</process>
</soap:Body>
</soap:Envelope>
```

Fig 2. SOAP message structure of invocation.

In the last phase (phase 5) is implemented the logic WS Gateway that allows the machine receiving SOAP requests on any of the operations of the service

that it offers, and translate them into signals understood by the machine itself. Once the request is received determines whether an activity or process is. And in the last case it triggers a BPEL engine running the related algorithm.

The advantage of this dynamic is that the elements of architecture are independent and could be implemented in any language and run on different platforms. Just they make sure they use the standards defined: SOAP, WSDL and UDDI.

4 Implementation of the platform intelligent management of industrial machine

The implementation and integration of the whole platform is organized into three sections, following the scheme that was presented in Fig. 1, the first one contains the parameters of design and implementation of the WS Gateway, the second one details the server used for registration and the third one focuses on the system of central management of the BPMS platform.

4.1 WS Gateway

The WS Gateway is responsible for publishing through the network several services which are offered by the machine to which it is connected. In addition to the WS Gateway incorporate an engine of execution of BPEL core processes. Providing, in this way, full functionality and flexibility to industrial machine. In the next subsection we present the details of implementation of WS Gateway and his physical and logical levels.

4.1.1 WS Gateway Physical Platform

In order to implement the WS Gateway we have selected a generic embedded device that is called XPORT of Lantronix brand [17], that device has several characteristics that contribute to the development of the platform. The Xport is an embedded web server that has the characteristics of a microcomputer that allows the redirection of data from Ethernet port to serial port[17].

This device has been selected for several reasons, such as: low cost, high-speed response, Ethernet-series gateway, handling C standard language for his programming, ability to handle various network protocols such as TCP, UDP, DHCP, etc.

To carry out our prototype of the platform has been used a industrial simulation scenario belonging to Staudinger GMHB [18] Fig 3, which deployed a number of drivers of the ICPDAS brand [19]. specifically the ICPDAS 7055D for managing the actuators in each subescenario either a conveyor belt, an industrial warehouse, etc.

In order to industrial simulation scenario we've implemented a interface that we've called WS Gateway which uses the embedded device XPORT to exchange messages with elements that control the various sensors and actuators. So that, the scenario operates with standard bus RS485, therefore we've provided this characteristic to the embedded device XPORT. The main features of standard RS485 are his possible transmission of data over long distances and the use of models in automatic control type master / slave.

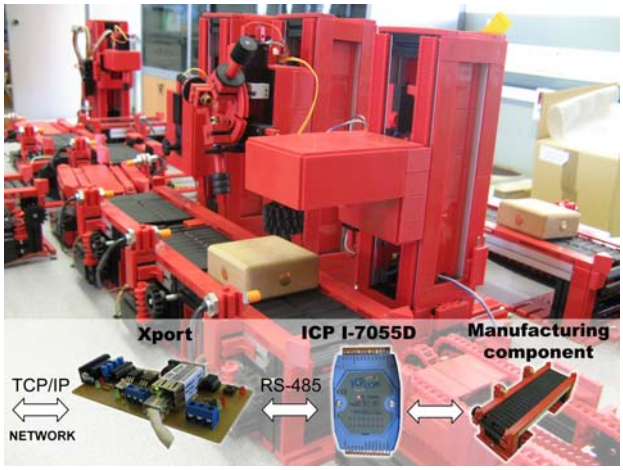

Fig 3. Industrial Simulation Scenario.

The model of embedded device XPORT that was used doesn't have this standard but the embedded device had the standard RS232 that can be easily coupled using an integrated circuit MAX485, which is managed by pins control XPORT. There are many embedded systems like the XPORT that offers several industrial standards [17] as Modbus, or even the same RS485. But a system can handle both standards (RS232 and RS485) generates a high added value, to configure and programming the embedded device and also to manage the industrial machines.

The WS Gateway is able to manage the embedded system and also create the bridge between the management system and Web modules that are part of ICPDAS industrial simulation scenario.

4.1.2 UDDI Register on the XPORT

In the WS Gateway we've implemented a module capable of register the services offered by each machine through SOAP messages on the server UDDI. To get this goal we've used the TCP functions which provide the operating system of XPORT device.

Also to validate the implementation of the proposal is used an example which defines a service with a series of activities and processes like shows the Table 1.

Table 1. Activities and Processes.

ACTIVITIES
Start to left or to right.
Stop.
Get the state presence sensor

PROCESSES
Move the conveyor belt until The object is in the sensor Presence.
Move the object until the sensor detects his presence.

Also we've defined for each one of the processes a BPEL document which has references to the WSDL interface across the Partnerlinks.

On the other hand, and referring to aspects of correspondence between WSDL / BPEL and UDDI in [15] we can establish a sequence of actions that must follow in order to register a service in a UDDI registry:

1. Get a token to UDDI registration, this will enable us to authenticate and perform all operations following publication.

2. Register the portType of the WSDL interface as a UDDI tModel. From portType we get the kind of information, local name, namespace and location of the WSDL document that defines the portType.

3. Register each one of the processes that presents the service as a tModel where is indicated the information of name's process, location of BPEL document that describes it and tModel reference to the portType that have previously registered.

4. Register the binding of the WSDL interface as a tModel of UDDI, indicating the type of information entity name, location of the file containing the WSDL binding, referring to portType that implements this, his protocol and transport.

5. Register the element Service of WSDL as an entity of UDDI businessService. We'll get the information type of entity, name, namespace and the list of ports that the service supports.

6. Register WSDL ports as an entity businessTemplate of UDDI. We get information that portType and binding this port implements, the local name and location of the service.

In this way, once completed the six steps above the service provided by the new machine connected to the network would be registered and also it would be available to accept petitions on any of its activities or processes.

On the other hand and continue with the design of the platform to perform compositions of several services was necessary to develop a small execution BPEL engine which it would be embedded within the WS Gateway and this is responsible for the interpretation of this language for modeling processes, this is follows in the following paragraph.

4.1.3 BPEL Engine embedded on XPORT

Another module that we have implemented within the system embedded WS Gateway, is a basic engine of BPEL language that allows interpret and execute the demands on the activities and processes of the machine.

In order to make this possible, WS Gateway must be able to interpret a SOAP message that contains a BPEL document which receives it through the Ethernet port by a specific TCP port.

Once the device has interpreted the BPEL document and extracted operations to be carried out by the machine, WS Gateway communicates with industrial machinery and sends commands according to the operations required by the user.

In order to do this, it has been necessary identify the type of messages that should be interpreted; studying the grammar of BPEL language, getting through all of their possibilities a subset functional that allow show results quickly and finally, study the API industrial machinery to achieve execute the transactions described by BPEL processes.

The Messages that BPEL engine must interpret are SOAP requests which are encapsulated BPEL processes.

In Fig 4 is shown the BPEL's grammar selected for this first deployment. The grammar is based on the implementation of tasks (activities or processes) in sequence, ignoring for the moment the other possibilities of language. We propose as part of future work, incorporating more complex grammatical structures as conditionals, loops, simultaneous execution, and so on.

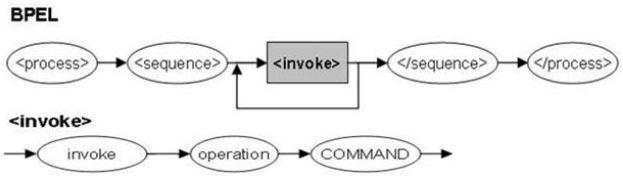

Fig 4. Basic BPEL's Grammar.

COMMAND has a corresponded with any atomic activities of each machine or a predefined process for them. In the case of the conveyor belt, the possible commands can be any of those defined in Table 1.

Finally, we have studied the API machines, identifying its atomic activities and some basic predefined processes for each machine, which are composed of several atomic activities. It also identified the correspondence between the commands and signals to be sent from WS Gateway towards the

lowest level in the machine. For example, to the conveyor belt, the command "Stop", was sent as a chain *"@0100 \ n"*.

4.2 UDDI Server

As a container for the available platform services was used an UDDI server. Specifically Apache jUDDI (juddi09rc4 version), which is an implementation of UDDI server developed in Java. This server is responsible for maintaining the services that sends the WS Gateway about the industrial machinery and provide an interface to search for activities and processes for that services.

4.3 Basic BPMS Prototype

BPMS designed implements some elements or phases of the life cycle of BPM modelling and execution. Central management System or BPMS consists of three sections: UDDI Manager, Business Processes Editor and Orchestration System, which are detailed below.

4.3.1 UDDI manager and Business processes Editor

In this module, as a first approximation, we have develop a Java application that accesses the UDDI registry and get a list of all registered services and operations that presents each of them. We have used UDDI4J library to interact with the UDDI registry and WSDL4J library for storing the WSDL structures WSDL.

If we indicate the URL where the UDDI server is located and click on Search button, we get an ordered tree with the different registered services into the UDDI registry, as well as the various activities and processes that they offer.

This module has another feature that is the composition of a comprehensive process that take place the machinery that had previously registered their services. The current developed prototype has implemented a simplification of this editor that allows only incorporate references to activities and processes previously registered into the UDDI registry, rather than predefined processes designed by the company.

Thus, from the services list the business processes editor will generate a sequence of activities and processes to compose this overall process Fig 5.

4.3.2 Orchestration System

For the orchestration system we have designed a first approximation where the processes planner used is a simplification that allows only invoke activities and processes without passing a custom code. A subsequent extension of the project should incorporate the logic allowing generate requests to incorporate more sophisticated BPEL code. For the process execution we have used the Apache library WSIF (Web Services Invocation Framework).

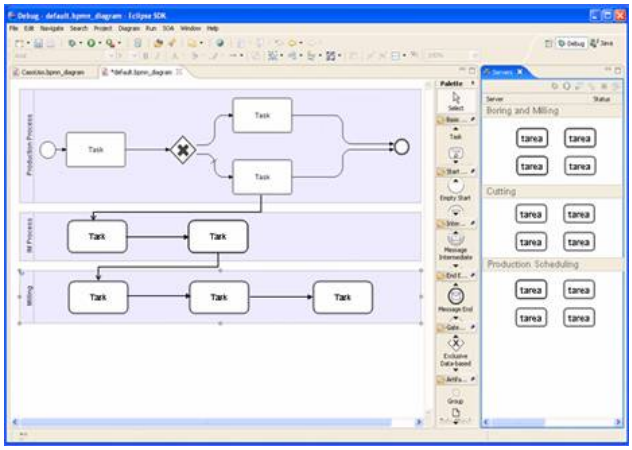

Fig 5. New global process composition.

5 Test

In order to the system tests we have used the mentioned scenario of the company Staudinger GMHB, which is connected to the WS Gateway.

Subsequently, in a different computer has been installed the jUDDI server, in which are registered the services supplied by the machinery. Then through the Java developed application are obtained the services provided by the machinery and with these it is possible compose BPEL processes that make up the new global processes. In a later extension of the project this BPEL document will be executed by the orchestration system described in the preceding paragraphs. Under current project, from the list of operations are generated SOAP messages with BPEL code that invokes different activities and processes.

These messages are sent to WS Gateway and when it receives them, the BPEL engine implemented locally interprets the message and executes the requested actions.

In our case the BPEL process contains operations for a conveyor belt, where they can carry out activities such as starting the belt and stop the belt, or processes as carrying the belt until the next sensor or take the belt until the end of its trail.

6 Conclusion

In this article we have developed a management platform for industrial machinery. The main novelty is that the functionality of the machinery will be displayed as processes exposed as Web services, which, individually, provide functions that contribute to achieving an overall process or activity.

The use of embedded devices in the industrial machinery environment allows that the industry converges towards IT, because most of its processes are away from web environments and even more of business models.

Using IT emerging paradigms like UDDI, SOAP, BPEL we will address the requirements of the new production models like mass customization that require flexible and dynamic management models to adapt to change.

The platform developed involves a mechanism for integration of industrial machinery to the increasingly complex production models.

Looking to the future development we pretend to incorporate more knowledge about the processes definition through the use of ontologies and semantic web, which generates greater intelligence for the transparent interaction of all parties that make up the environment for industrial machinery and business.

References:

[1] Gilart-Iglesias V, Maciá-Perez F, Jorquera-Marcos D, Mora-Gimeno F, Industrial Machines as a service: Modelling insdutrial Machinery process. *5th IEEE International Conference on Industrial Informatics (INDIN'07)*, pp 737-742 Viena Austria,2007.

[2] Branch, M., B. Bradley, and IEEE. Real-time web-based system monitoring. *Conference Record of 2006 Annual Pulp and Paper Industry Technical Conference*, pp. 133-136, 2006.

[3] Herrera L. Telemetría y telegestión en procesos industriales mediante canales inalámbricos Wi Fi utilizando instrumentación virtual y dispositivos PDA (Personal Digital Assitant). *III Jornadas para el Desarrollo de Grandes Aplicaciones de Red*, pp 119-129 Alicante, España. 2006.

[4] Gilart-Iglesias V, Maciá-Perez F, Hernández A, Jorquera D, Chamizo J. A model for developing J2EE applications based design patterns, *Proceedings of IADIS International Conferences on apllied computing.* pp 352-360 Algarve, Portugal, 2005.

[5] Jorquera, D, Gilart V, Maciá F. Modelo de reconfiguración dinámica de los elementos de producción orientado a la fabricación flexible *IV jornadas para el desarrollo de grandes aplicaciones de red*, pp 15-32. Alicante España, OFP Yeclagrafic, 2007.

[6] Topp, U, Müller P, Web based service for embedded devices. *International Workshop on Web Services: Research Standardization and deployment (WS-RSD'02)*, Lecture Notes in computer science Web Service and Database Systems, pp. 141-153, 2003.

[7] *SOAP specification*. [cited 2008 May 20], Available from: http://www.w3.org/TR/soap/.

[8] *The ITEA Initiative.* [cited 2008 May 20], Available from: http://www.itea-office.org/.

[9] *The SIRENA project.*[cited 2008 May 20], Available from: http://www.sirena-itea.org/Sirena/Home.htm.

[10] *Especificación WSDL del W3C.* [cited 2008 May 20], Available from: http://www.w3.org/TR/wsdl.

[11] *UDDI Data Structure Refrence.* [cited 2008 May 8]; Available from: http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm.

[12] *BPEL specification.* [cited 2008 May 20], Available from: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

[13] Newcomer E. *Understanding Web Services – XML, WSDL. SOAP, and UDDI.* Addison Wesley. 2002

[14] *XML specification.* [cited 2008 May 20]; Available from: http://www.w3.org/XML/.

[15] *Using BPEL4WS in a UDDI registry.* [cited 2008 May 20], Available from: http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-bpel.htm.

[16] *Using WSDL in a UDDI registry*. [cited 2008 May 20], Available from: http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm.

[17] Lantronix. *Embedded Device Servers Xport.* [cited 2008 July 10], Available from: http://www.lantronix.com/device-networking/embedded-device-servers/xport.html

[18] *Staudinger GMBH.* [cited 2008 June 15], Available from: http://www.staudinger-est.de/intl/en/simulation02.aspx?group=1.

[19] ICPDAS. [cited 2008 June 18], Available from: http://www.icpdas.com/.